Contemporary Programming & Software Design

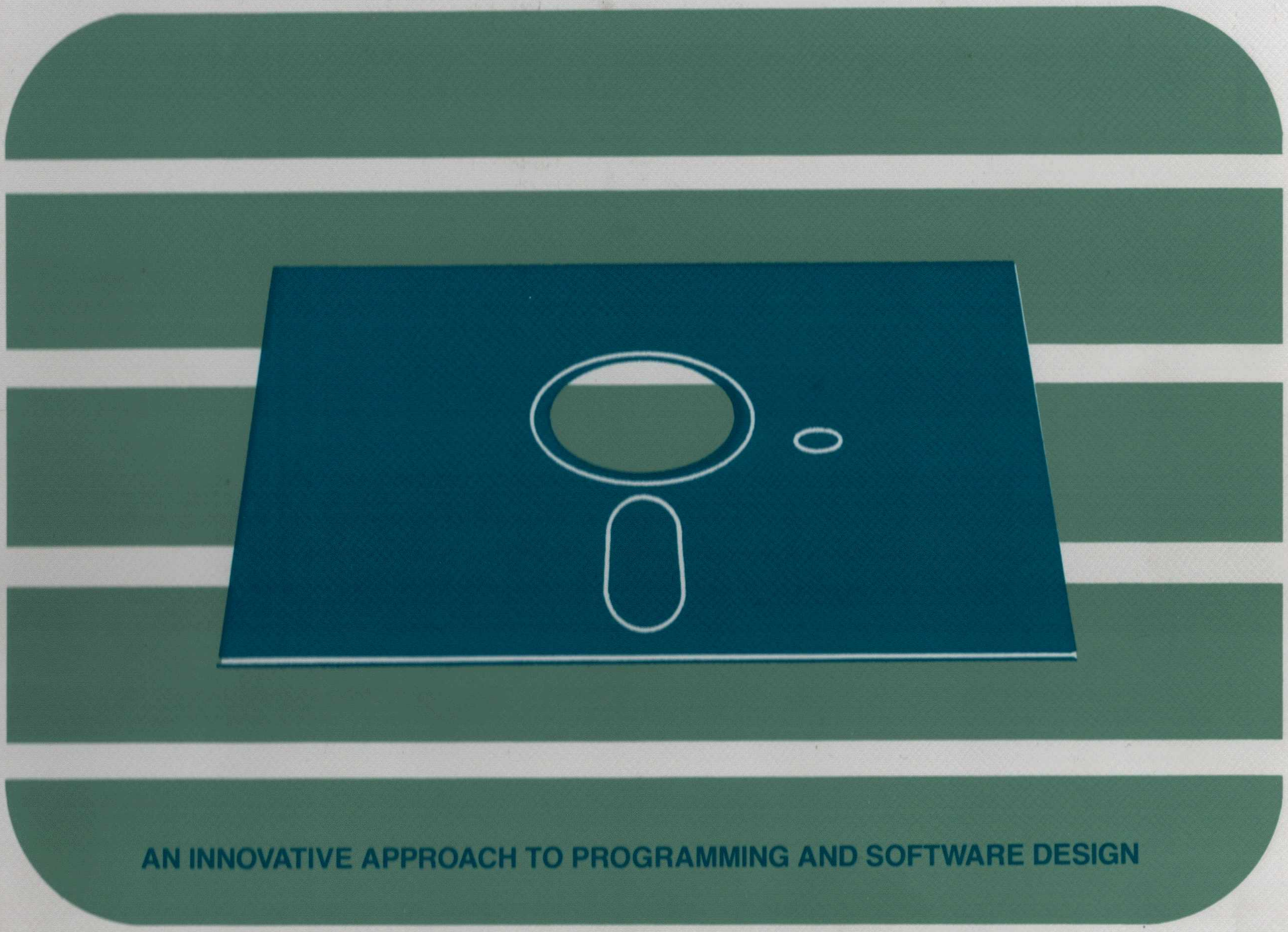
AN INNOVATIVE APPROACH
TO PROGRAMMING
AND SOFTWARE DESIGN

GETTING STARTED IN PROGRAMMING & SOFTWARE DESIGN
ATTACKING THE PROBLEM
HOW TO DESIGN THE SOLUTION AND ARRANGE IT LOGICALLY
CODING THE PROGRAM HIGH-LEVEL LANGUAGES
CODING THE PROGRAM ASSEMBLY LANGUAGES

McGraw Hill

Contemporary Programming & Software Design

McGraw Hill



AN INNOVATIVE APPROACH TO PROGRAMMING AND SOFTWARE DESIGN

Contemporary Programming & Software Design

McGraw Hill
3939 Wisconsin Avenue
Washington, D.C. 20016

Telephone 202/244-1600

WELCOME TO THE WORLD OF CONTEMPORARY PROGRAMMING AND SOFTWARE DESIGN!

We are delighted at your decision to participate in McGraw-Hill's exciting new Contemporary Programming and Software Design Series. We are confident that you will find the Series to be a unique and stimulating approach to acquiring the skills necessary to make your software - and your personal computer - do specifically what you want them to do.

The key objective of this series is to teach you how to create your own software - either from scratch or by making key modifications to the programs you now have. You'll be guided through the process of building computer programs step-by-step with each module you receive. Each of the ten modules will contain interactive hands-on instruction, and you'll have numerous opportunities to work with the types of computer programs used every day in many homes and businesses.

This first module provides an introductory overview of the programming procedures. You will discover how to identify the primary programming steps. To reinforce your learning, you can perform hands-on corrections and extensions of an actual declining interest loan amortization program. (This is the same type of program used to calculate mortgage payments, car loans, and personal loans.) By the time you have completed this first module, you will be well on your way to understanding more about Contemporary Programming and Software Design.

Your next module in the series will be Defining and Attacking the Problem. You will learn how to formulate the basic problem statement, a critical first step in the programming process. You will experiment with rearranging this problem statement into the actual tasks that must be performed by the computer. As in all modules in the Contemporary Programming and Software Design Series, you'll have the opportunity to work with both written instructional materials and the interactive diskettes to provide a lively hands-on learning experience.

We invite you now to begin Module 1, and to see for yourself just how rewarding and interesting Contemporary Programming and Software Design can be.

Sincerely,

Kenneth J Bigelow

Kenneth J. Bigelow
Project Engineer

Contemporary
Programming &
Software Design

McGraw-Hill
3939 Wisconsin Avenue
Washington, D.C. 20016

Telephone 202/244-1600

CONTEMPORARY PROGRAMMING & SOFTWARE DESIGN COURSE OVERVIEW

The Contemporary Programming & Software Design Series is an exciting new approach to programming instruction. This series emphasizes the generic nature of the programming process. Unlike many classroom or seminar-type courses which pertain to just one language or one machine, this series highlights those basic principles that apply to all languages and all applications.

Here is a brief description of the ten modules you'll receive with the Contemporary Programming & Software Design Series:

Module 1: Getting Started in Programming & Software Design. This module provides an introduction and general overview of the topics of programming and software design. Included are definitions of terms, an overview of the programming process, and a look at software design procedures. The interactive diskette contains a declining interest loan amortization program that you can analyze and modify.

Module 2: Defining and Attacking the Problem. In this module, you'll learn the crucial first step in programming: creation of a clear, precise problem definition. The module will then lead you into discovering various ways to attack or define the problem-solving process. The coordinated diskette will include a checkbook register program that you will use to help resolve some everyday problem situations.

Module 3: How to Design the Solution and Arrange It Logically. Now that you have learned how to define and attack the problem, you are ready to learn how to design the solution. This module will guide you through the procedures of logically sequencing tasks and diagramming a solution to the problem.

Module 4: Coding the Program: High-Level Languages. In this module, you'll learn more about the differences — and the amazing similarities — between the vast number of programming languages. You will learn more about languages such as BASIC, Fortran, COBOL, and other high-level languages. At this stage you will have the skills to begin running programs of your own.

Module 5: Coding the Program: Assembly Language. In this module, you'll have the opportunity to learn more about what your computer is actually doing. Learning more about assembly language will make you more familiar with your microcomputer, and you'll more fully understand the distinction between assembly language and high-level languages.

(over)

Module 6: Testing and Debugging Made Easy. This module illustrates a number of techniques for locating, identifying, and correcting potential errors in a computer program. Even commercial programmers spend a considerable amount of time fine-tuning their programs. You'll find that the techniques you learn in this module will prove to be significant timesavers for your future computer-related projects.

Module 7: Creating Meaningful Documentation. Documentation is an area that is often overlooked in the programming process. This module will discuss the reasons why clear and complete documentation is such an important programming step. Different types of documentation are discussed, as well as recommended applications. The few hours you spend working on this module may help to save you many hours of time in the future.

Module 8: Modifying and Updating Existing Programs. The need for modification often occurs after a long program is originally developed. Obviously, modification is often simpler than developing an entire new program. In this module, you'll learn to develop procedures to accomplish these modifications on many different levels.

Module 9: Tools and Tricks for Program Design. Here is where you'll receive some "inside information" from programming professionals. You'll learn about structured programming, modular techniques, and general programming standards that will give you an advantage in all your programming work.

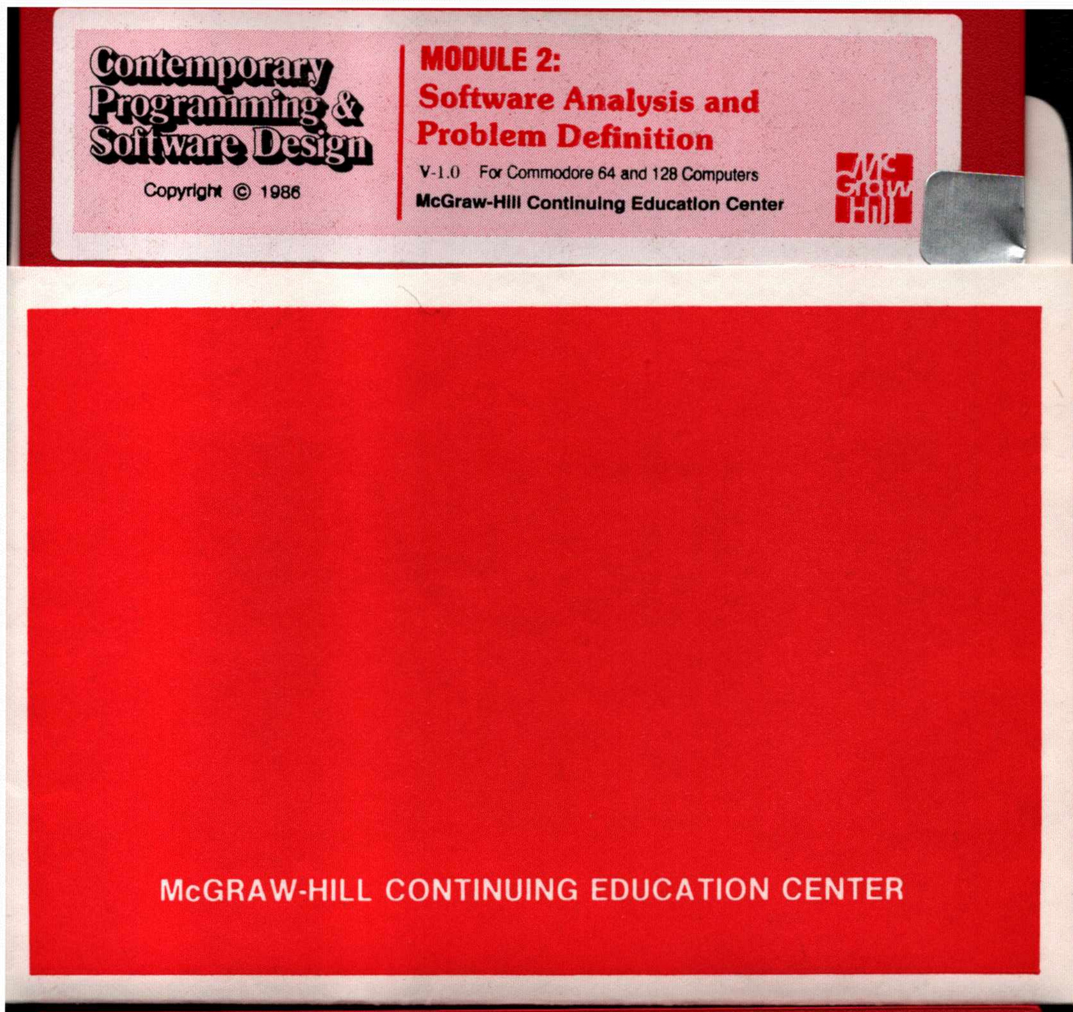
Module 10: Writing Advanced Programs. This final module discusses some practical advanced programming techniques. You'll learn more about the concepts of structured programming, and how to redesign programs to conserve memory space and improve execution speed. These techniques will have numerous applications in the software you'll be working with in the future.

As you can see, the Contemporary Programming & Software Design Series is an exciting, cumulative approach to learning more about programming and software design.

The interactive programs and texts you'll work with in every module are excellent learning tools in themselves. In addition, by the time you complete the series, you will have modified and combined these individual programs onto one master disk, ultimately creating your own financial planning package. This family financial planner will help you balance your budget, calculate loan amortization, and perform a multitude of other real-world applications that will make your personal financial planning more productive and satisfying.

As your own specific needs change over time, you'll be able to further modify your financial package to fit your current budgetary situations. You'll be surprised at how much you can do with the skills you have acquired through completing this exciting series.

** We reserve the right to modify module titles and/or sequence during the development or subsequent revision of the Contemporary Programming & Software Design Series.*



Contemporary Programming & Software Design

Copyright © 1986

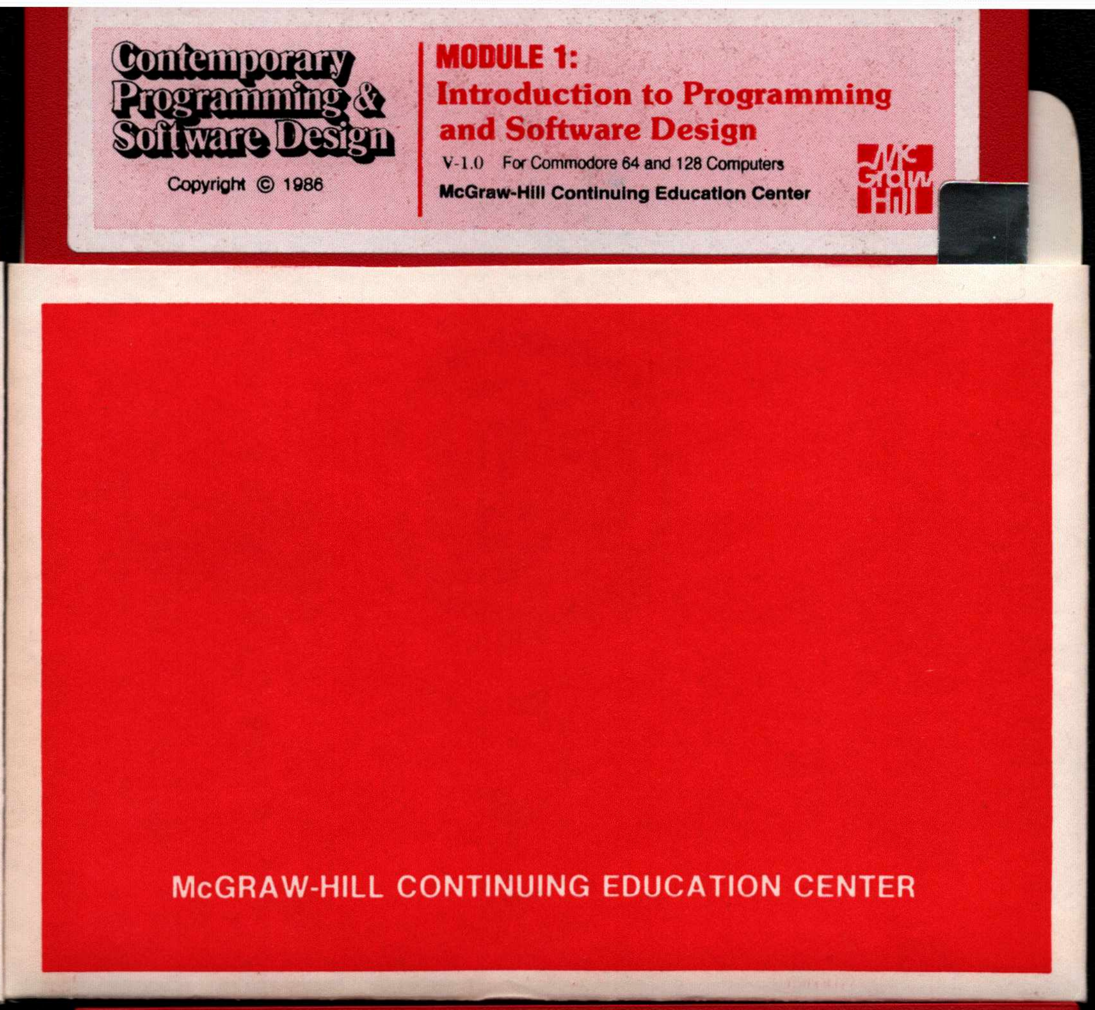
MODULE 2:

Software Analysis and Problem Definition

V-1.0 For Commodore 64 and 128 Computers

McGraw-Hill Continuing Education Center

McGRAW-HILL CONTINUING EDUCATION CENTER



Contemporary Programming & Software Design

Copyright © 1986

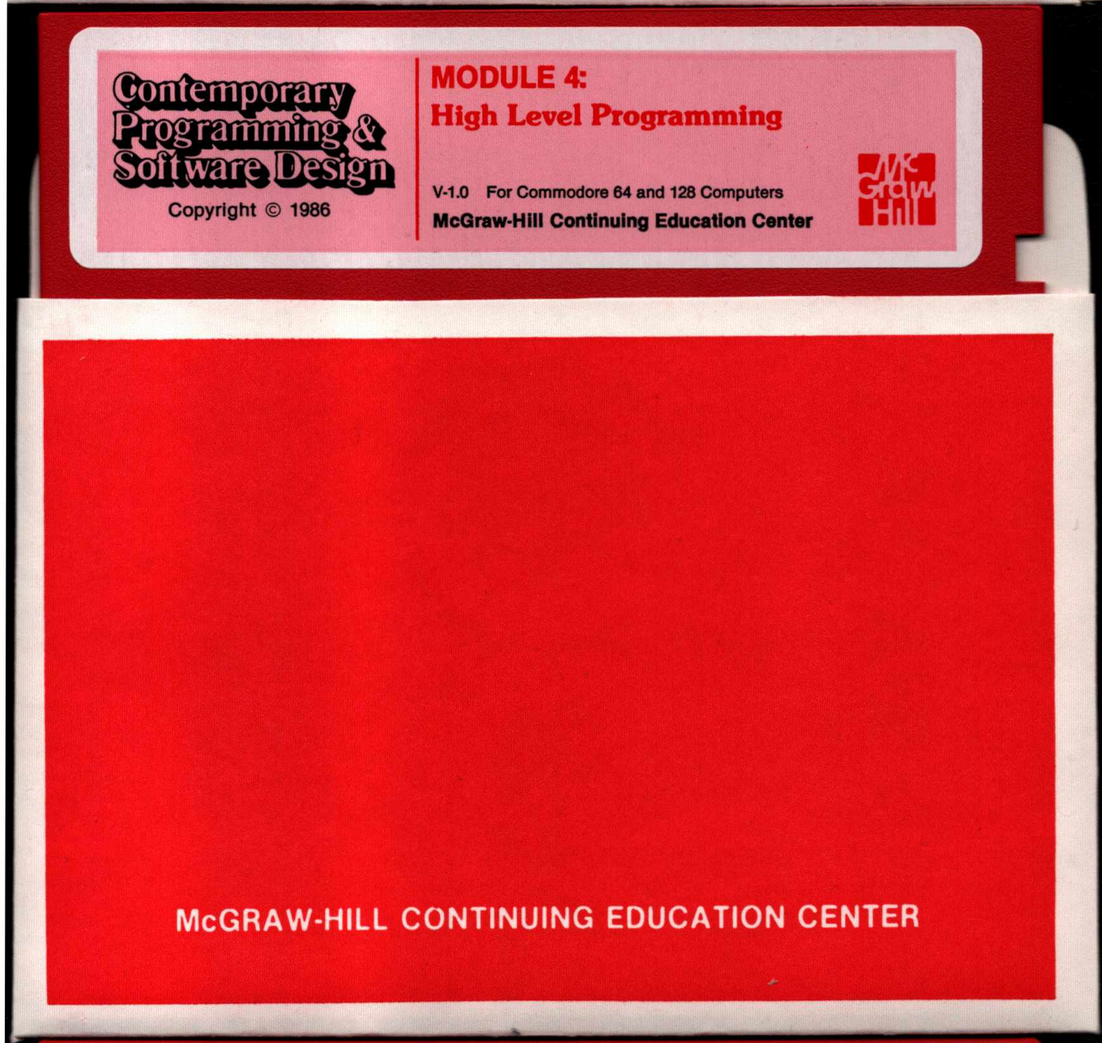
MODULE 1:

Introduction to Programming and Software Design

V-1.0 For Commodore 64 and 128 Computers

McGraw-Hill Continuing Education Center

McGRAW-HILL CONTINUING EDUCATION CENTER



Contemporary Programming & Software Design

Copyright © 1986

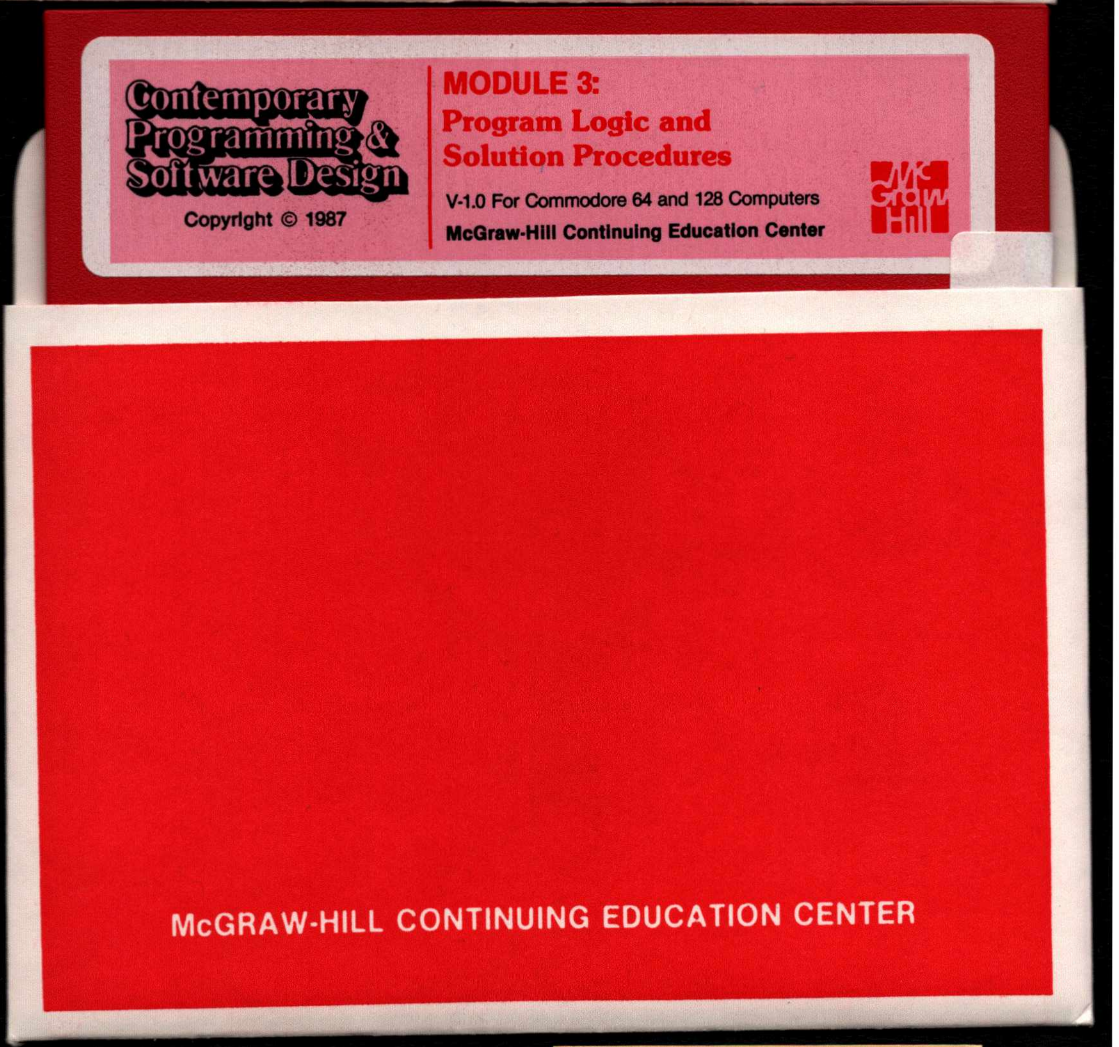
MODULE 4:

High Level Programming

V-1.0 For Commodore 64 and 128 Computers

McGraw-Hill Continuing Education Center

McGRAW-HILL CONTINUING EDUCATION CENTER



Contemporary Programming & Software Design

Copyright © 1987

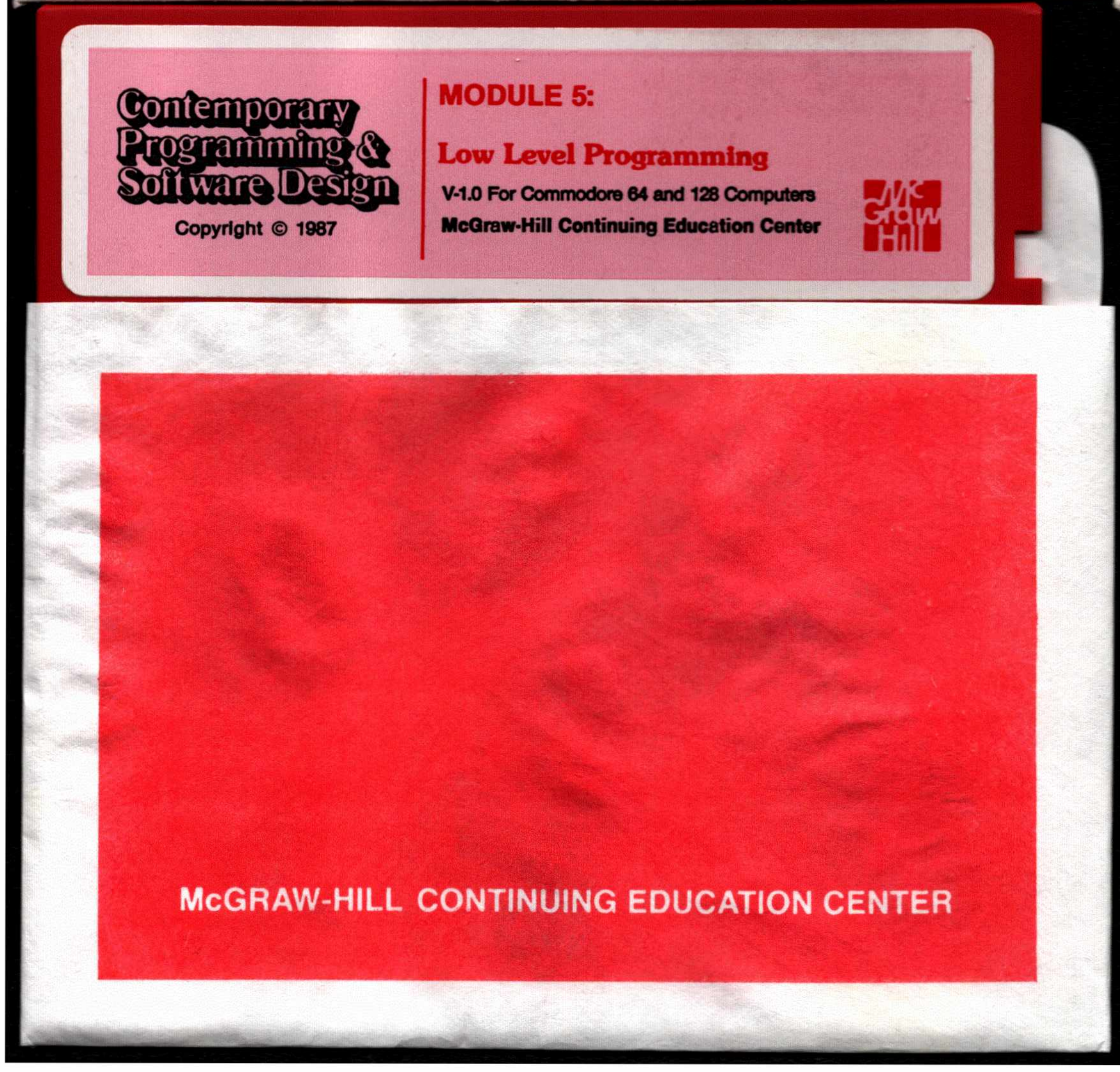
MODULE 3:

Program Logic and Solution Procedures

V-1.0 For Commodore 64 and 128 Computers

McGraw-Hill Continuing Education Center

McGRAW-HILL CONTINUING EDUCATION CENTER



Contemporary Programming & Software Design

Copyright © 1987

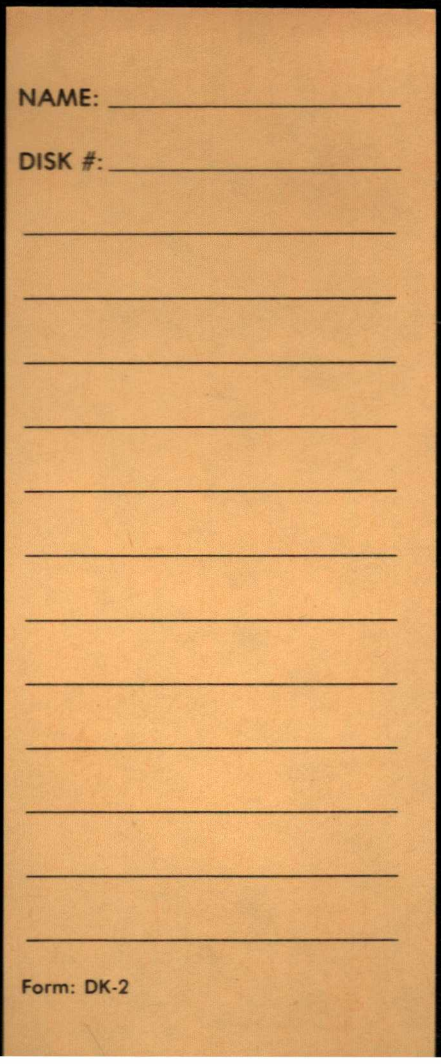
MODULE 5:

Low Level Programming

V-1.0 For Commodore 64 and 128 Computers

McGraw-Hill Continuing Education Center

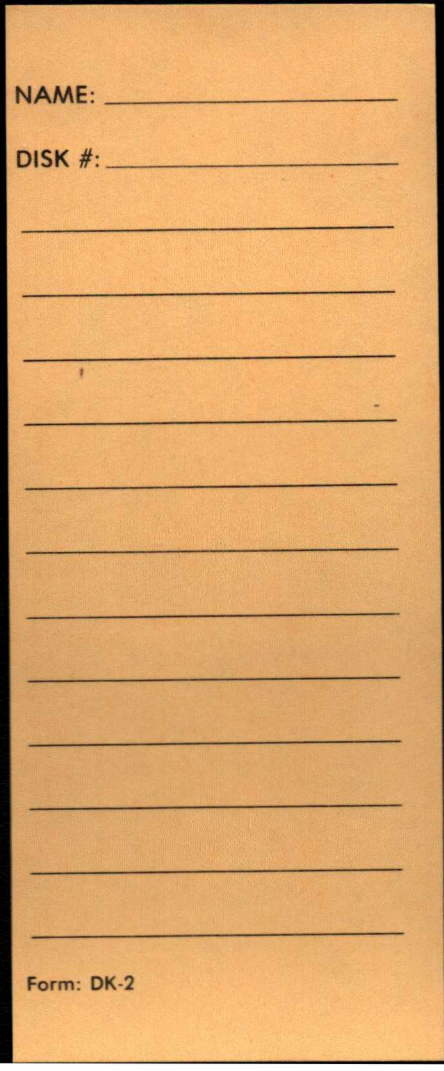
McGRAW-HILL CONTINUING EDUCATION CENTER



NAME:

DISK #:

Form: DK-2



NAME:

DISK #:

Form: DK-2

PROGRAM

BY

LANGUAGE

PAGE OF

DATE

A	B	C	D	E	F	G	H	I	J

McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue
Washington, D.C. 20016

Name ______________________________

Address ______________________________

City ____________________ **State** ____________ **Zip** __________

The above is used as a mailing label, please print in ink.

REPLACEMENT PARTS REQUEST

Use this card if parts to your module are missing, damaged, or defective. Be sure to return this card within 10 days after you receive the module.

Module Number ______________

Module Title ______________

____

Insert entire card in an envelope and mail to:

Customer Service — Parts Replacement
McGraw-Hill Continuing Education Center
3939 Wisconsin Avenue
Washington, D.C. 20016

Part No.	Description	Quantity

BF 132(707)